

Wt and DA

@Slip-1

Q.1)Write a PHP script to keep track of number of times the webpage has been accessed (Use Session Tracking).

Ans:

<?php

Session_start();

```
if(isset($_SESSION['pcount'])){
    $_SESSION['pcount']++;
}else{
    $_SESSION['pcount']=1;
}
```

Echo "You have visited this page " . \$_SESSION['pcount'] . " Time(s).";

?>

Q.2) Create 'Position_Salaries' Dataset. Build a linear regression model by identifying independent and Target variable. Split the variables into training and testing sets. Then divide the training and testing sets into a 7:3 ratio, respectively, and print them. Build a simple linear regression model.

Ans:

Import numpy as np

Import pandas as pd

From sklearn.model_selection import train_test_split

From sklearn.linear_model import LinearRegression

Create the Position_Salaries dataset

Data = {'Position': ['CEO', 'charman', 'director', 'Senior Manager', 'Junior Manager', 'Intern'],

'Level': [1, 2, 3, 4, 5, 6],



Edit with WPS Office

'Salary': [50000, 80000, 110000, 150000, 200000, 250000]]

Df = pd.DataFrame(data)

Identify the independent and target variables

X = df.iloc[:, 1:2].values

Y = df.iloc[:, 2].values

Split the variables into training and testing sets with a 7:3 ratio

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

Print the training and testing sets

Print("X_train:\n", X_train)

Print("y_train:\n", y_train)

Print("X_test:\n", X_test)

Print("y_test:\n", y_test)

Build a simple linear regression model

Regressor = LinearRegression() Regressor.fit(X_train, y_train)

Print the coefficients and intercept

Print("Coefficients:", Regressor.coef_)

Print("Intercept:", Regressor.intercept_)

@Slip-2

Q.1) Write a PHP script to change the preferences of your webpage like font style, font size, font color, background color using cookies. Displays elected settings on next webpage and actual implementation (with new settings) on third page (Use Cookies).

Ans:



Edit with WPS Office

Fristpage.html

<!DOCTYPE html>

<html>

<head>

<title>Change preferences</title>

</head> <body>

<h1>Change preferences</h1>

<form action="secondpage.php" method="post">

<label for="fontstyle">Font Style:</label>

<select name="fontstyle" id="fontstyle">

<option value="Arial">Arial</option>

<option value="Times New Roman">Times New Roman</option>

<option value="Verdana">Verdana</option>

</select>

<label for="fontsize">Font Size:</label>

<select name="fontsize" id="fontsize">

<option value="12">12</option>

<option value="14">14</option>

<option value="16">16</option>

</select>

<label for="fontcolor">Font Color:</label>

<input type="color" name="fontcolor" id="fontcolor">

<label for="bgcolor">Background Color:</label>

<input type="color" name="bgcolor" id="bgcolor">

<input type="submit" name="submit" value="Save">

</form>

</body>

</html>

Secondpage.php

<?php

\$username = \$_POST['username'];

\$password = \$_POST['password'];

// Set correct username and password

\$correct_username = 'myusername';

\$correct_password = 'mypassword';

// Check if entered username and password are correct

if (\$username == \$correct_username && \$password == \$correct_password) {

// Set session variable to mark user as logged in \$_SESSION['loggedin'] = true;

// Redirect user to welcome page

Header('Location: welcome.php');

Exit;

else {

// Decrement login attempts

if (isset(\$_SESSION['attempts'])) {

\$_SESSION['attempts']--;

else {

\$_SESSION['attempts'] = 3;

}

// Display error message if maximum login attempts exceeded

if (\$_SESSION['attempts'] <= 0) {

Echo "Maximum login attempts exceeded. Please try again later.";

else {

// Display error message

Echo "Invalid username or password. You have " . \$_SESSION['attempts'] . " Attempt(s) left.";

}

}

?>

if (isset(\$_POST['submit'])) {

\$fontstyle = \$_POST['fontstyle'];

\$fontsize = \$_POST['fontsize'];

\$fontcolor = \$_POST['fontcolor'];

\$bgcolor = \$_POST['bgcolor'];

// Set the cookie values

Setcookie('fontstyle', \$fontstyle, time() + 86400);

Setcookie('fontsize', \$fontsize, time() + 86400);

Setcookie('fontcolor', \$fontcolor, time() + 86400);

Setcookie('bgcolor', \$bgcolor, time() + 86400);

// Redirect to the next page

Header('Location: thirdpage.php');

Exit();

}

?>

Thirdpage.php

<?php

// Retrieve the cookie values

\$fontstyle = isset(\$_COOKIE['fontstyle']) ? \$_COOKIE['fontstyle'] : 'Arial';

\$fontsize = isset(\$_COOKIE['fontsize']) ? \$_COOKIE['fontsize'] : '12';

\$fontcolor = isset(\$_COOKIE['fontcolor']) ? \$_COOKIE['fontcolor'] : '#000000';

\$bgcolor = isset(\$_COOKIE['bgcolor']) ? \$_COOKIE['bgcolor'] : '#FFFFFF';

?>

<!DOCTYPE html>

<html>

<head>

<title>Page with new settings</title>

<style type="text/css">

Body {



Edit with WPS Office

Font-family:<?php echo \$fontstyle?>;

Font-size:<?php echo \$fontsize?>px;

Color:<?php echo \$fontcolor?>;

Background-color:<?php echo \$bgcolor?>;

}

</style>

</head> <body>

<h1>Page with new settings</h1>

<p> This is the page with the new settings. The font style is <?php echo \$fontstyle?>, the font size is <?php echo \$fontsize?>px, the font color is <?php echo \$fontcolor?>, and the background color is <?php echo \$bgcolor?>.</p>

</body>

</html>

Q.2) Create 'Salary' Dataset. Build a linear regression model by identifying independent and target

variable. Split the variables into training and testing sets and print them. Build a simple linear regression

model for predicting purchases.

Ans:

Import numpy as np

Import pandas as pd

From sklearn.model_selection import train_test_split

From sklearn.linear_model import LinearRegression

Create the Salary dataset

Data = {'YearsExperience': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

'Salary': [50000, 60000, 70000, 80000, 90000, 100000, 110000, 120000, 130000, 140000]}

Df = pd.DataFrame(data)

Identify the independent and target variables

X = df.iloc[:, 0:1].values

Y = df.iloc[:, 1].values



Edit with WPS Office

Split the variables into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

Print the training and testing sets

Print("X_train:\n", X_train)

Print("y_train:\n", y_train)

Print("X_test:\n", X_test)

Print("y_test:\n", y_test)

Build a simple linear regression model

Regressor = LinearRegression() Regressor.fit(X_train, y_train)

Print the coefficients and intercept

Print("Coefficients:", Regressor.coef_)

Print("Intercept:", Regressor.intercept_)

@Slip-3

Q.1) Write a PHP script to accept username and password. If the first three chances, username and password entered is correct then display a message "Welcome message" otherwise display an error message (Use Session).

Ans:

<?php

// Start session

Session_start();

// Check if login form has been submitted

if (isset(\$_POST['submit'])) {

// Get username and password input from user



Edit with WPS Office



Edit with WPS Office



Edit with WPS Office

```
#Predictsingleobservation
Observations=[[0,30,87000]]
Prediction=Lr.predict(observation)
Print(prediction)

#Predictmultipleobservations
Observations=[[0,30,87000],[1,50,45000],[1,22,30000]]
Predictions=Lr.predict(observations)
Print(predictions)
```

@Slip-4

Q.1]WriteaPHPscripttoacceptEmployeeDetails(Eno,Ename,Address)onfirstpage.On second Pageacceptearning(Basic,DA,HRA).OnthirdpageprintEmployeeinformation(Eno,Ename, Address, Basic,DA,HRA,Total)[[UseSession]

Ans:

Firstpage.php

```
<?php
Session_start();

?>

<IDOCTYPEhtml>
<html>
<head>
<title>EmployeeDetails</title>
</head> <body>
```

W

Edit with WPS Office

```
<labelform="eno">EmployeeNo:</label>

```

```
</form>
</body>
</html>
```

<?php

```
//StoreemployeeDetailsinession
If(isset($_POST['eno'])&&isset($_POST['ename'])&&isset($_POST['address']))
{
$_SESSION['eno']= $_POST['eno'];
$_SESSION['ename']= $_POST['ename'];
$_SESSION['address']= $_POST['address'];
}
?>
```

Secondpage.php

```
<?php
Session_start();

?>

<IDOCTYPEhtml>
<html>
<head>
<title>Earnings</title>
</head><body>
<h1>Earnings</h1>
<formmethod="POST" action="thirdpage.php">
```

W

Edit with WPS Office

```
<labelform="basic">Basic:</label>

```

```
</form>
</body>
</html>
```

<?php

```
//Storeearningsinession
If(isset($_POST['basic'])&&isset($_POST['da'])&&isset($_POST['hra']))
{
$_SESSION['basic']= $_POST['basic'];
$_SESSION['da']= $_POST['da'];
$_SESSION['hra']= $_POST['hra'];
}
?>
```

Thirdpage.php
<?php
Session_start();

```
//Calculatetotalearnings
$total= $_SESSION['basic']+ $_SESSION['da']+ $_SESSION['hra'];
?>
```

```
<IDOCTYPEhtml>
<html>
<head>
<title>EmployeeInformation</title>
```

W

Edit with WPS Office

```
</head> <body>
<h1>EmployeeInformation</h1>
<p><strong>EmployeeNo:</strong><?phpecho$_SESSION['eno'];?></p>
<p><strong>EmployeeName:</strong><?phpecho$_SESSION['ename'];?></p>
<p><strong>Address:</strong><?phpecho$_SESSION['address'];?></p>
<p><strong>Basic:</strong><?phpecho$_SESSION['basic'];?></p>
<p><strong>DA:</strong><?phpecho$_SESSION['da'];?></p>
<p><strong>HRA:</strong><?phpecho$_SESSION['hra'];?></p>
<p><strong>TotalEarnings:</strong><?phpecho$total;?></p>
</body>
</html>
```

Q.2]BuildasimplelinearegressionmodelforFishSpeciesWeightPrediction.
Ans:

Importandasasp
Importrandom
Fromsklearn.linear_modelImportLinearRegression

#createthedataset
Fish_species=['Tuna','Salmon','Trout','Bass','Sardine','Cod','Mackerel']
Weights=[]

```
Forinrange(50):
    Fish_weight=[]
    Forinrange(7):
        Weight=random.randint(1,20)
        Fish_weight.append(weight)
    Weights.append(fish_weight)
```

Df=pd.DataFrame(weights,columns=fish_species)

W

Edit with WPS Office

```
#createthelinearregressionmodel
X=df.iloc[:,1]#Independentvariables
Y=df.iloc[:,1]#targetvariable
```

Model=LinearRegression() Model.fit(X,y)

```
#Predicttheweightofanewfishspecies
New_fish=[[10,12,15,7,4,8]]#exampleinput
Predicted_weight=model.predict(new_fish)
Print("Predictedweight:",predicted_weight)
```

@Slip-5
Q.1]CreateXMLfilenamed"Item.xml"withitem-name,item-rate,itemquantitystorethedetails of5 Items ofdifferentTypes.

Ans:

```
Item.xml
<items>
<itemtype="Electronics">
<name>Television</name>
<rate>500</rate>
<quantity>10</quantity>
</item>
<itemtype="Clothing">
<name>Shirt</name>
<rate>50</rate>
<quantity>20</quantity>
</item>
<itemtype="Grocery"> <name>Rice</name>
<rate>40</rate>
<quantity>30</quantity>
```

W

Edit with WPS Office

```
</item>
<itemtype="Books">
<name>HarryPotterandthePhilosopher'sStone</name>
<rate>20</rate>
<quantity>50</quantity>
</item>
<itemtype="Sports">
<name>Football</name>
<rate>100</rate>
<quantity>5</quantity>
</item></items>
```

Q.2]Usetheirdataset. WriteaPythonprogramtoviewsomebasicstatisticaldetailslike percentile, Mean,stdetc.Othreespeciesof'Iris-setosa','Iris-versicolor'and'Iris-virginica'.Applylogistic regression OnthedatasettoidentifydifferentSpecies(setosa,versicolor,verginica)ofIrisflowersgiven just4 Features.sepalandpetallengthsandwidths.Findtheaccuracyofthemodel.

Ans:

```
Importpandasasp
Fromsklearn.datasetsimportload_iris
Fromsklearn.linear_modelImportLogisticRegression
Fromsklearn.model_selectionimporttrain_test_split
Fromsklearn.metricsimportaccuracy_score
```

#loadtheirdataset Iris=load_iris()

```
#createtheDataFramefromthedataset
Df=pd.DataFrame(Iris.data,columns=Iris.feature_names)
Df['target']=Iris.target
#viewbasicstatisticaldetailsofthreedifferentSpecies
Print("StatisticaldetailsofIris-setosa:")
Print(dff[dff['target']==0].describe())
```

W

Edit with WPS Office

```
Print("StatisticaldetailsofIris-versicolor:")
Print(dff[dff['target']==1].describe())

Print("StatisticaldetailsofIris-virginica:")
Print(dff[dff['target']==2].describe())

#splithedatointotrainandtestingsets
X=df.iloc[:,1:]
Y=df.iloc[:,1]
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

```
#fitalogisticregressionmodel
Logreg=LogisticRegression() Logreg.fit(X_train,y_train)
```

#makepredictionsonthetestset Y_pred=logreg.predict(X_test)

```
#calculatetheaccuracyofthemodel
Accuracy=accuracy_score(y_test,y_pred)
Print("Accuracyofthelogisticregressionmodel:",accuracy)
```

@Slip-6
Q.1]WritePHPscripttoread"book.xml"fileintosimpleXMLObject.Displayattributesand elements.
(simple_xml_load_file()function)

Ans:

```
<?php
//LoadtheXMLfileintoaSimpleXMLObject $xml=simplexml_load_file("book.xml");
```

```
//DisplaytheattributesandelementsoftheSimpleXMLObject
Echo"Bookattributes:<br>;
```

W

Edit with WPS Office

```
Echo"ISBN:".$xml['isbn'];<br>;
Echo"Publisher:".$xml['publisher'];<br>; Echo"<br>;
```

```
Echo"Bookelements:<br>;
Echo"Title:".$xml->title."<br>;
Echo"Author:".$xml->author."<br>;
Echo"Description:".$xml->description."<br>;
?>
```

```
Book.xmlfile
<?xmlversion="1.0"?>
<bookisbn="978-3-16-148410-0"publisher="ExamplePublisher">
<title>ExampleBook</title>
<author>JohnDoe</author>
<description>Thisisanexamplebook.</description>
</book>
```

Q.2]Createthefollowingdatasetinpython&Convertthecategoricalvaluesintonumeric format.Apply Theapriorialgorithmontheabovedatasettogeneratethefrequentitemsetsandassociation rules.Repeat Treproccsswithdifferentmin_supvalues.
TID=[1,1]:"bread","milk",1,2["bread","diaper","beer","eggs",1,3["milk","diaper","beer","coke",1,4=["bread","milk","diaper","beer",1,5=["bread","milk","diaper","coke"]]

```
Ans:
Importpandasasp
Frommlxtend.preprocessingimportTransactionEncoder
Frommlxtend.frequent_patternsimportapriori,association_rules
```

#createthedataset
TID=

W

Edit with WPS Office

```
{1:["bread","milk"],2:["bread","diaper","beer","eggs"],3:["milk","diaper","beer","coke"],4:["bread","mil
k","diaper","beer"],5:["bread","milk","diaper","coke"]}
Transactions=[]

forkey,valueinTID.items():
    Transactions.append(value)

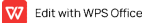
#convertthecategoricalvaluesintonumericformat
Te=TransactionEncoder()
Te_ary=te.fit_transform(transactions)
Df=pd.DataFrame(te_ary.columns+te.columns_) #applythepriorialgorithminwithdifferentmin_supvalues
Min_sup_values=[0.2,0.4,0.6]
Formin_supinmin_sup_values:
    Frequent_itemsets=apriori(df,min_support=min_sup_use,colnames=True)
    Rules=association_rules(frequent_itemsets,metric="confidence",min_threshold=0.7)
    Print("Min_sup:",min_sup)
    Print("FrequentItemsets:")
    Print(frequent_itemsets)
    Print("AssociationRules:")
    Print(rules)

@Slip-7

Q.1)WriteaPHPscripttoread"Movie.xml"fileandprintallMovieTitleandActorNameoffile using
OMDocumentParser."Movie.xml"fileshouldcontainfollowinginformationwith atleast5 records
Withvalues.MVieInfoMovieNo,MovieTitle,ActorName,ReleaseYear.
Ans:

Phpfile

<?php
//LoadtheXMLfile
```



Edit with WPS Office

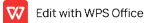
```
$xml=newDOMDocument();
$xml->load('Movie.xml');
//Getallthemovie nodes
$movies=$xml->getElementsByTagName("MovieInfo");

//Loopthroughachmovie nodeandprintthemovie titleandactorname

foreach($moviesas$movie){
    Echo"MovieTitle:".$movie->getElementsByTagName("MovieTitle")[0]->textContent."<br>";
    Echo"ActorName:".$movie->getElementsByTagName("ActorName")[0]->textContent."<br>";
}
?>

XMLfile

<?xmlversion="1.0"?>
<MovieList>
  <MovieInfo>
    <MovieNo>1</MovieNo>
    <MovieTitle>TheShawshankRedemption</MovieTitle>
    <ActorName>TimRobbins</ActorName>
    <ReleaseYear>1994</ReleaseYear>
  </MovieInfo>
  <MovieInfo>
    <MovieNo>2</MovieNo>
    <MovieTitle>TheGodfather</MovieTitle>
    <ActorName>MarlonBrando</ActorName>
    <ReleaseYear>1972</ReleaseYear>
  </MovieInfo>
  <MovieInfo>
    <MovieNo>3</MovieNo>
    <MovieTitle>TheDarknight</MovieTitle>
```



Edit with WPS Office

```
<ActorName>ChristianBale</ActorName>
<ReleaseYear>2008</ReleaseYear>
</MovieInfo>
<MovieInfo>
  <MovieNo>4</MovieNo>
  <MovieTitle>TheGodfather:PartII</MovieTitle>
  <ActorName>AlPacino</ActorName>
  <ReleaseYear>1974</ReleaseYear>
</MovieInfo>
<MovieInfo>
  <MovieNo>5</MovieNo>
  <MovieTitle>12AngryMen</MovieTitle>
  <ActorName>Henryfonda</ActorName>
  <ReleaseYear>1957</ReleaseYear>
</MovieInfo>
</MovieList>
```

Q.2)DownloadtheMarketbasketdataset.Writeapythonprogramtoreadthedata setand displays
Information.Preprocessthedata(dropnullvaluesetc.)Convertthecategoricalvaluesinto numeric
Format.Applythepriorialgorithmontheabovedatasettogeneratethefrequentitemsetsand association Rules..

```
Ans:
Importpandasaspd
frommlxtend.preprocessingimportTransactionEncoder
frommlxtend.frequent_patternsimportapriori,association_rules
```

```
#readthedata set
Df=pd.read_csv("Market_Basket_Optimisation.csv",header=None)
```

```
#dropnullvalues
Df.dropna(inplace=True)
```

```
}

//Returntrueifbothusernameandpasswordarevalid Returntrue;

}
```

Q.2)Createyourowntransactiondatasetandapplytheaboveprocessonyourdataset.
Ans:

Items=["item1','item2','item3','item4"]

```
Transactions=[["item1','item2','item3'],
               ["item2','item3'],
               ["item1','item2','item4'],
               ["item1','item4'],
               ["item2','item3','item4'],
               ["item1','item3','item4'],
               ["item1','item2'],
               ["item1','item3'],
               ["item3','item4'],
               ["item2','item4']
               ]
```

```
frommlxtend.preprocessingimportTransactionEncoder
frommlxtend.frequent_patternsimportapriori,association_rules #Convertthetransactionsintoabinarymatrix
Te=TransactionEncoder()
Te_ary=te.fit_transform(transactions)
```

```
#ConvertthebinarymatrixintoapandasDataFrame
Df=pd.DataFrame(te_ary.columns+te.columns_)
```

#Generatefrequentitemsetswithanimumsupportof0.3

```
#convertcategoricalvaluestonumericusingone-hotencoding
Te=TransactionEncoder()
Te_ary=te.fit(df.values).transform(df.values)
Df=pd.DataFrame(te_ary.columns+te.columns_)
```

#generatefrequentitemsetsusingapriorialgorithm
Frequent_itemsets=apriori(df,min_support=0.01,use_colnames=True)

#generateassociationrulesfromfrequentitemsets
Rules=association_rules(frequent_itemsets,metric="lift",min_threshold=1)

```
#displayinformation
Print("OriginalDataset:\n")
Print(df.head())
Print("\nFrequentItemsets:\n")
Print(frequent_itemsets)
Print("\nAssociationRules:\n")
Print(rules)
@Slip-8
```

Q.1)WriteaJavaScripttodisplaymessage"Examsarenear,haveyoustartedpreparingfor?" {useAlert
Box}andAcceptanytwonumbersfromuseranddisplayadditionoftwonumber.(UsePromptand Confirmbox) AAAns:
//Displaymessageusingalertbox
Alert("Examsarenear,haveyoustartedpreparingfor?");

```
//Accepttwonumbersfromuserusingpromptandconfirmboxes
Letnum1=prompt("Enterfirstnumber:");
Letnum2=prompt("Entersecondnumber:");
LetconfirmMsg='Areyoureadytowanttoadd$(num1)and$(num2)?';
```

```
//Showconfirmationmessagetouserusingconfirmbox
LetconfirmResult=confirm(confirmMsg);
```

Frequent_itemsets=apriori(df,min_support=0.3,use_colnames=True)

#Generateassociationruleswithanimumconfidenceof0.7
Association_rules=association_rules(frequent_itemsets,metric="confidence",min_threshold=0.7)

#Printthefrequentitemsetsandassociationrules
Print(frequent_itemsets)
Print(association_rules)

@Slip-10

Q.1)CreateaHTMLfiletoinserttextbeforeandafteraParagraphusingQuery.(Hint:Use before() Andafter()).

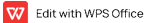
Ans:

```
<!DOCTYPEhtml>
<html>
<head>
  <title>inserttextbeforeandafterapagraphusingQuery</title>
  <scriptsrc="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head> <body>

  <h1>inserttextbeforeandafterapagraphusingQuery</h1>

  <p>Thisisapagraph.</p>

  <script>
    $(document).ready(function(){
      $("p").before("Textinsertedbeforethepagraph.");
      $("p").after("Textinsertedafterthepagraph."); });
    </script>
  </body>
```



Edit with WPS Office

</html>

Q2) Create the following dataset in python & Convert the categorical values into numeric format. Apply The apriori algorithm on the above dataset to generate the frequent itemsets and association rules. Repeat The process with different min_sup values.
TID=1["eggs","milk","bread"],2=["eggs","apple"],3=["milk","bread"],4=["apple","milk"],5=["milk","apple","bread"]
Ans:

```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
```

```
# Create the dataset
Dataset = [
    1: ["eggs", "milk", "bread"],
    2: ["eggs", "apple"],
    3: ["milk", "bread"],
    4: ["apple", "milk"],
    5: ["milk", "apple", "bread"]
]
```

```
# Convert categorical values into numeric format
Te = TransactionEncoder()
Te_ary = Te.fit(dataset.values()).transform(dataset.values())
Df = pd.DataFrame(Te_ary.columns, Te.columns_)
```

```
# Apply Apriori algorithm to generate frequent itemsets and association rules
Min_sup = 0.4
Frequent_itemsets = apriori(Df, min_support=Min_sup, use_colnames=True)
Association_rules = association_rules(Frequent_itemsets, metric="confidence", min_threshold=0.6)
```

 Edit with WPS Office


```
# Print the frequent itemsets and association rules
Print("\n Association Rules:\n", association_rules)
```

@Slip-11

Q.1) Write a JavaScript program to accept the name of a student, change the font color to red, font size to 18 if the student's name is present; otherwise, on clicking on the empty text box, display an image which changes its size (Use onblur, onload, onmouseover, onmouseleave, onmouseover, onmouseleave, onmouseover, onmouseleave).

```
<!DOCTYPE html>
<html>
<head>
<title> JavaScript Example </title>
<style>
#name {
    font-size: 14px;
    color: black;
}
</style>
</head>
<body>
<input type="text" id="name" onblur="changeStyle()" onmouseover="changeSize()"
onmouseout="resetSize()" onmousedown="changeColor()" onmouseup="resetColor()">

</body>
</html>
<script>
function changeStyle() {
    let name = document.getElementById("name").value;
    if (name) {
        document.getElementById("name").style.fontSize = "18px";
        document.getElementById("name").style.color = "red";
    } else {
        document.getElementById("img").style.display = "block";
    }
}
```

 Edit with WPS Office

```
}

function changeSize() {
    document.getElementById("name").style.fontSize = "16px";
}


function resetSize() {
    document.getElementById("name").style.fontSize = "14px";
}

function changeColor() {
    document.getElementById("name").style.color = "blue";
}

function resetColor() {
    document.getElementById("name").style.color = "red";
}

function changeImageSize() {
    document.getElementById("img").style.width = "200px";
    document.getElementById("img").style.height = "200px";
}
</script>
</body>
</html>
```

Q2) Create the above dataset in python & Convert the categorical values into numeric format. Apply The apriori algorithm on the above dataset to generate the frequent itemsets and association rules. Repeat The process with different min_sup values.

 Edit with WPS Office

```
TID=1["butter","bread","milk"],2=["butter","flour","milk","sugar"],3=["butter","eggs","milk","salt"],4=["eggs"],5=["butter","flour","milk","salt"]]
```

Ans:


```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
```

```
# Create the dataset
Dataset = [
    ["butter", "bread", "milk"],
    ["butter", "flour", "milk", "sugar"],
    ["butter", "eggs", "milk", "salt"],
    ["eggs"],
    ["butter", "flour", "milk", "salt"]
]
Df = pd.DataFrame(Dataset)
```

```
# Convert the categorical values into numeric format
Te = TransactionEncoder()
Te_ary = Te.fit(Dataset).transform(Dataset)
Df = pd.DataFrame(Te_ary.columns, Te.columns_)
```

```
# Generating frequent itemsets using Apriori algorithm with different min_sup values
Min_sup_values = [0.4, 0.3, 0.2]
Formin_sup_in_min_sup_values:
Frequent_itemsets = apriori(Df, min_support=Min_sup, use_colnames=True)
Print("Frequent itemsets with minimum support:", min_sup)
Print(frequent_itemsets)
```


```
# Generating association rules
Rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
Print("Association Rules with minimum support:", min_sup)
Print(rules)
```

 Edit with WPS Office


@Slip-12

Q.1) Write an Ajax program to read contact data file and print the contents of the file in a tabular format. When the user clicks on the print button, the contact data file should contain s.no, name, residence number, mobile number, address. [Enter at least 3 records in the contact data file].

```
Ans:
Html file
<<!DOCTYPE html>
<html>
<head>
<title> Contact List </title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="script.js"></script>
</head>
<body>
<button id="printBtn"> Print Contacts </button>
<br>
<table id="contactTable">
<thead>
<tr>
<th> Sr.No. </th>
<th> Name </th>
<th> Residence Number </th>
<th> Mobile Number </th>
<th> Address </th>
</tr>
</thead>
<tbody>
<!-- Contact list will be displayed here -->
</tbody>
</html>
```

 Edit with WPS Office

```
</table>
</body>
</html>
Ajax file
$(document).ready(function() {
    // Event listener for print button
    $("#printBtn").click(function() {
        // Ajax request to read contact data file
        $.ajax({
            url: "contact.dat",
            dataType: "text",
            success: function(data) {
                // Split the file contents into lines
                var lines = data.split("\n");
                // Iterate over each line and create a table row
                var tableRows = "";
                for (var i = 0; i < lines.length; i++) {
                    var columns = lines[i].split(",");
                    if (columns.length == 5) { // Only process valid rows
                        tableRows += "<tr>";
                        for (var j = 0; j < columns.length; j++) {
                            tableRows += "<td>" + columns[j] + "</td>";
                        }
                        tableRows += "</tr>";
                    }
                }
                // Add the table rows to the table body
                $("#contactTable tbody").html(tableRows);
            },
            error: function(jqXHR, textStatus, errorThrown) {
                alert("Error: " + errorThrown);
            }
        });
    });
});
```

 Edit with WPS Office

```
});
});

Q.2) Create 'heights-and-weights' Dataset. Build a linear regression model by identifying independent and target variable. Split the variables into training and testing sets and print them. Build a simple linear regression model for predicting purchases.
```

Ans:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```


```
# Create a random dataset with 10 samples
heights = np.random.normal(170, 10, 10)
weights = np.random.normal(70, 5, 10)

# Combine the two arrays into a single dataset
Dataset = pd.DataFrame({'Height': heights, 'Weight': weights})
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(Dataset['Height'], Dataset['Weight'], test_size=0.2, random_state=42)

# Create a linear regression model and fit it to the training data
lr_model = LinearRegression()
lr_model.fit(X_train.values.reshape(-1, 1), y_train)

# Print the model coefficients
Print("Model Coefficients: ", lr_model.coef_)

# Predict the weights for the test data and print the predictions
Y_pred = lr_model.predict(X_test.values.reshape(-1, 1))
```

 Edit with WPS Office

Print("Predictions: ", y_pred)


@Slip-13

Q.1) Write an Ajax program where the user is requested to write his or her name in a text box, and the server keeps sending back responses while the user is typing. If the user's name is not entered, then the message displayed will be, "Stranger, please tell me your name!". If the name is Rohit, Virat, Dhoni, Ashwin or Harbhajan, the server responds with "Hello, master!". If the name is anything else, the message will be, "I don't know you!".

Ans:
Html file

```
<!DOCTYPE html>
<html>
<head>
<title> AJAX Program </title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</head>
<body>
<label for="name"> Enter your name: </label>
<input type="text" id="name" name="name">
<div id="response"> </div>
<script src="ajax.js"></script>
</body>
</html>
```

Ajax file

 Edit with WPS Office

```
$(document).ready(function(){
    //Attachaneventlistener tothenameinputfield $(`#name`) on`input`,function(){

        //Getthenameenteredbytheuser Varname=$(this).val();

        //SendanAJAXrequesttotheserver
        $.ajax({
            url:'server.php', type:'POST',
            data:{name:name}, success:function(response){

                //Updatetheresponsesedwiththeserver'sresponse
                $('#response').html(response);

            }
        });
    });
});
```

Filename:Server.php

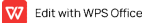
```
<?php

//Getthenameenteredbytheuser
$name=$_POST['name'];

//Checkifthenameisempty
if(empty($name)){
    Echo`Stranger,pleasetellmeyourname!`;
}

//Checkifthenameisoneofthemasernames
elseif($name=='Rohit' || $name=='Virat' || $name=='Dhoni' || $name=='Ashwin' || $name== 'Harbhajan'){
    Echo`Hello,master!`;
}

}
```



Edit with WPS Office

```
//Otherwise,theserverdoesn'tknowtheuser
Else{
    Echo$name.`!don'tknowyou!`;
}

Q.2)DownloadnurserydatasetfromUCI.Buildalinearregressionmodelbyidentifying independent
Andtargetvariable.Splitthevariablesintotrainingandtestingsetsandprinthem.Builda simplelinear
Regressionmodelforpredictingpurchases.
Ans:

Importpandasaspd
Importnumpyasnp
Fromsklearn.model_selectionimporttrain_test_split
Fromsklearn.linear_modelimportLinearRegression
```

#Loadthedataset

```
url=https://archive.ics.uci.edu/ml/machine-learning-databases/nursery/nursery.data
names=['parents','has_nurs','form','children','housing','finance','social','health','class']
dataset=pd.read_csv(url,name=names)
```

```
#Identifyindependentandtargetvariables
X=dataset.drop('class',axis=1)
Y=dataset['class']

#Convertcategoricalvariablesintonumericalvariablesusingone-hotencoding
X=pd.get_dummies(X)
#Splittintotrainingandtestingsets
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

#Buildalinearregressionmodel

Model=LinearRegression() Model.fit(X_train,y_train)

```
#Printthecoefficientsofthemodel
Print("Intercept: ",model.intercept_)
Print("Coefficients: ",model.coef_)
```

```
#Predictthetargetvariableforthetestingset
Y_pred=model.predict(X_test)
```

```
#EvaluatethemodelusingMeanSquaredError(MSE)
Mse=np.mean((y_test-y_pred)**2)
Print("MSE: ",mse)
```

@Slip-14

Q.1)CreateTEACHERtableasfollowsTEACHER(tno,name,qualification,salary).WriteAjax
Programtoselectateachersnameandprintheselectedteachersdetails.

AAns:

```
Jsfile
<!DOCTYPEhtml>
<html>
<head>

<title>TeacherDetails</title>

<scriptsrc=https://code.jquery.com/jquery-3.6.0.min.js></script>

</head> <body>

<selectid="teacher-list">

    <optionvalue="">SelectTeacher--</option>
    <optionvalue="1">JohnDoe</option>
    <optionvalue="2">JaneSmith</option>
    <optionvalue="3">BobJohnson</option>
```

```
</select>
<buttonid="submit-btn">GetDetails</button>
<divid="details"></div>

<script>
    $(document).ready(function(){

        $(`#submit-btn`).click(function(){

            Vartno=$(`#teacher-list`).val();

            if(tno==""){
                Alert("Pleasetelectateacher.");
                Return;
            }

            $.ajax({ url:'teacherdetails.php', method:'POST',
                data:{tno:tno}, success:function(response){

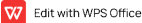
                    $('#details').html(response);

                },
                Error:function(xhr,status,error){
                    Console.log(xhr.responseText);
                }
            });

        });
    });
</script>
</body>
</html>
```

Phpfileteacherdetails.php

```
<?php
//Connecttodatabase
$servername="localhost";
$username="username";
$password="password";
```



Edit with WPS Office

```
$dbname="database_name";
$conn=mysqli_connect($servername,$username,$password,$dbname);
```

```
//Checkconnection
if($conn){
    Die("Connectionfailed: ".mysqli_connect_error());
}
}
```

```
//Retrieveselectedeacherdetails
if(isset($_POST['tno'])){
    $tno=$_POST['tno'];
    $sql="SELECT*FROMTEACHERWHEREtno=$tno"; $result=mysqli_query($conn,$sql);
```

```
if(mysqli_num_rows($result)>0){
    $row=mysqli_fetch_assoc($result);
    Echo`TeacherName:`, $row['name'];<br>;
    Echo`Qualification:`, $row['qualification'];<br>;
    Echo`Salary:`, $row['salary'];<br>;
}
else{
    Echo`Nodatafound.`;
}
}
```

```
//Closedatabaseconnection
mysqli_close($conn);
?>
```

Q.2)Createthefollowingdatasetinpython&Convertthecategoricalvaluesintonumeric format.Apply
Theapriorialgorithmontheabovedatasettogeneratethefrequentitemsetsandassociation rules.Repeat
Theprocesswithdifferentmin_sup_values.

```
TID=[1:["apple","mango","banana"],2:["mango","banana",
"cabbage","carrots"],3=["mango","banana","carrots"],4=["mango","carrots"]]AAns:
```

```
Frommlxtend.preprocessingimportTransactionEncoder
Frommlxtend.frequent_patternsimportapriori
#Createthedataset
TID=[1:["apple","mango","banana"],
2:["mango","banana","cabbage","carrots"],
3:["mango","banana","carrots"],
4:["mango","carrots"]]

#Convertthecategoricalvaluesintonumericformat
Te=TransactionEncoder()
Te_ary=te.fit([TID[i]foriinTID]).transform([TID[i]foriinTID]) Df=pd.DataFrame(te_ary.columns=te.columns_)
```

```
#Applytheapriorialgorithewithdifferentmin_supvalues
Min_sup_values=[0.25,0.5,0.75] Formin_sup,min_sup_values:

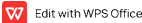
Frequent_itemsets=apriori(df,min_support=min_sup,use_colnames=True)
Print("Frequentitemsetswithin_min_sup=",min_sup)
Print(frequent_itemsets)
Print("\n")
```

@Slip-15

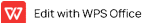
Q.1)WriteAjaxprogramtofetchsuggestionswhenuseristypingintextbox.(eglikegoogle
Suggestions.Hintcreatearrayofsuggestionsandmatchingstringwillbedisplayed).

```
Ans:
<!DOCTYPEhtml>
<html>
<head>
```

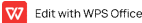
<title>AJAXAutoSuggestionsExample</title>



Edit with WPS Office



Edit with WPS Office



Edit with WPS Office

```
No] Company] model] year
1. Tata. Nexon. 2017
2. MG. Astor. 2021
```

```
<script>
    FunctionfetchSuggestions(str){
        if(str.length==0){
            Document.getElementById("suggestions").innerHTML=""; Return;
        }

        Varsuggestions=["apple","banana","cherry","dates","elderberry","fig",
"grape","honeydew","kiwi","lemon"];
        Varmatches=[];
        For(vari=0;i<suggestions.length;i++){
            If(suggestions[i].toLowerCase().startsWith(str.toLowerCase())){
                Matches.push(suggestions[i]);
            }
        }
        If(matches.length>0){
            Document.getElementById("suggestions").innerHTML=
matches.join("<br>");
        }else{
            Document.getElementById("suggestions").innerHTML="No
suggestionsfound";
        }
    }
</script>
</head> <body>

<inputtype="text"onkeyup="fetchSuggestions(this.value)">
<divid="suggestions"></div>

</body>
</html>
Q.2)Createthefollowingdatasetinpython&Convertthecategoricalvaluesintonumeric format.Apply
Theapriorialgorithmontheabovedatasettogeneratethefrequentitemsetsandassociation rules.Repeat
Theprocesswithdifferentmin_supvalues.
```

```
3. Kia. Seltos. 2019
4. Hyundai. Creta. 2015
```

Ans:
Importpandasaspd

```
#Createthedataset
Data={ 'No':[1,2,3,4],
'Company': ['Tata','MG','Kia','Hyundai'],
'Model': ['Nexon','Astor','Seltos','Creta'],
'Year': [2017,2021,2019,2015]}
```

Df=pd.DataFrame(data)

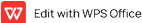
```
#Convertcategoricalvaluesintonumericformat
Df['Company']=pd.Categorical(Df['Company'])
Df['Model']=pd.Categorical(Df['Model'])
```

```
Df['Company']=df['Company'].cat.codes
Df['Model']=df['Model'].cat.codes
```

```
Print(df)
Frommlxtend.frequent_patternsimportapriori
Frommlxtend.frequent_patternsimportassociation_rules
```

```
#Generatefrequentitemsetswithin_min_sup=0.5
Frequent_itemsets=apriori(df,min_support=0.5,use_colnames=True)
Print(frequent_itemsets)
```

```
#Generateassociationruleswithin_threshold=0.7
Association_rules=association_rules(frequent_itemsets,metric="confidence", min_threshold=0.7)
Print(association_rules)
```



Edit with WPS Office


@Slip-16

Q.1)WriteAjaxprogramtogetbookdetailsfromXMLfilewhenusersselectabookname. CreateXML Fileforstoringdetailsofbook(title,author,year,price).

Ans:


Xmlfilebook_details.xml

```
<books>
  <book>
    <title>TheGreatGatsby</title>
    <author>F.ScottFitzgerald</author>
    <year>1925</year>
    <price>10.99</price>
  </book> <book>
    <title>ToKillAMockingbird</title>
    <author>HarperLee</author>
    <year>1960</year>
    <price>8.99</price>
  </book> <book>
    <title>1984</title>
    <author>GeorgeOrwell</author>
    <year>1949</year>
    <price>6.99</price>
  </book> <book>
    <title>PrideandPrejudice</title>
    <author>JaneAusten</author>
    <year>1813</year>
    <price>7.99</price>
  </book>
</books>
```

 Edit with WPS Office

Ajaxfile

```
<DOCTYPEhtml>
<html>
<head>
  <title>BookDetails</title>
  <scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script> <script>
    $(document).ready(function(){
      $("select").change(function(){
        Varbook=$($this).val();
        $.ajax({ url:"book_details.xml", dataType:"xml",
          success:function(xml){
            $(xml).find("book").each(function(){
              Vartitle=$($this).find("title").text();
              If(title==book){
                Varauthor=
                  Varyear=$($this).find("year").text();
                  Varprice=$($this).find("price").text();
                  $("#details").html("Author:"+author
                    );
                  });
                });
              });
            });
          </script>
        </head> <body>
          <select>
            <option>Selectabook</option>
            <option>TheGreatGatsby</option>
            <option>ToKillAMockingbird</option>
          </select>
        </body>
      </html>
```

 Edit with WPS Office

```
<option>1984</option>
<option>PrideandPrejudice</option>
</select>
<divid="details"></div>
</body>
</html>
```

Q2).Consideranytextparagraph.Preprocesstheextortoremoveanyspecialcharactersand digits. Generate thesummaryusingextractivesummarizationprocess.

Ans:


```
Importre
Importnltk
Fromnltk.corpusimportstopwords
Fromnltk.tokenizeimportsent_tokenize,word_tokenize
Fromheapqimportnlargest
```

#Sampletextparagraphyoucanwriteanytext

```
Text="Naturallanguageprocessing(NLP)isasubfieldoflinguistics,computerscience,
informationengineering,andartificialintelligenceconcernedwiththeinteractionsbetween
computersandhumanlanguages,particularlyhowtoprogramcomputerstoprocessand
analyzelargeamountsofnaturallanguagedata.Challengesinnaturallanguageprocessing
frequentlyinvolve speechrecognition,naturallanguageunderstanding,andnaturalanguage
generation.Thehistoryofnaturallanguageprocessinggenerallystartedinthe1950s,although
workcanbefoundfromearlierperiods."
#Removespecialcharactersanddigits
Text=re.sub("[^a-zA-Z]",'',text)
```

```
#Tokenizethetextintosentences
Sentences=sent_tokenize(text)
```

```
#Tokenizeeachsentenceintowordsandremovestopwords
Stop_words=set(stopwords.words('english'))
Words=[]
```

 Edit with WPS Office

```
Forsentenceinsentences:
  Words.extend(word_tokenize(sentence))
Words=[word.lower()forwordinwordsifword.lower()notinstop_words]
```

```
#Calculatewordfrequency
Word_freq=nlk.FreqDist(words)
```

```
#Calculatesentencescoresbasedonwordfrequency
Sentence_scores=[]
Forsentenceinsentences:
  Forwordinword_tokenize(sentence.lower()):
    Ifwordinword_freq:
      Iflen(sentence.split(" "))<30:
        Ifsentencenotinsentence_scores:
          Sentence_scores[sentence]=word_freq[word] Else:
            Sentence_scores[sentence]+=word_freq[word]
```

```
#Generatesummarybyselectingtop3sentenceswithhighestscores
Summary_sentences=nlargest(3,sentence_scores,key=sentence_scores.get)
Summary=""
join(summary_sentences)
```


Print(summary)

@Slip-17

Q.1)WriteaJavaScriptProgramtoshowHelloGoodMorningmessageonloadeventusing alertbox AnddisplaytheStudentregistrationfrom.

Ans:

```
<DOCTYPEhtml>
<html>
```

 Edit with WPS Office


```
<head>
  <title>StudentRegistrationForm</title> <script>
    Window.onload=function(){
      Alert("HelloGoodMorning!");
    };
  </script>
</head> <body>
```

```
<h1>StudentRegistrationForm</h1>
<form>
  <labelfor="name">Name:</label>
  <inputtype="text" id="name" name="name" required><br><br>
  <labelfor="email">Email:</label> <inputtype="email" id="email" name="email" required><br><br>
  <labelfor="phone">Phone:</label>
  <inputtype="tel" id="phone" name="phone" required><br><br>
  <labelfor="address">Address:</label>
  <textareaid="address" name="address" required></textarea><br><br>
  <inputtype="submit" value="Submit">
</form>
</body>
</html>
```

Q.2)Considertextparagraph.So,keepworking.Keepstriving.Nevergiveup.Falldownseventimes,getupeight.Ease isagreaterthreattoprogress thanhardship.Easeisagreaterthreattoprogress than Hardship.So,keepmoving,keepgrowing,keeplearning.Seeyoutowork.Preprocesstheexto remove Anyspecialcharactersanddigits. Generate thesummaryusingextractivesummarization process. Ans:

```
Importre
Fromnltk.tokenizeimportsent_tokenize
```

#Textparagraph

 Edit with WPS Office

Text="So,keepworking.Keepstriving.Nevergiveup.Falldownseventimes,getupeight.Ease isagreaterthreattoprogress thanhardship.Easeisagreaterthreattoprogress thanhardship. So,keepmoving,keepgrowing,keeplearning.Seeyoutowork."

```
#Removespecialcharactersanddigits
Text=re.sub("[^A-Za-z]",'',text)
#Tokenizethesentences
Sentences=sent_tokenize(text)
```

```
#Calculatescoreofeachsentencebasedonthenumberofwords
#Thesentenceswithmorewordswillhaveahigherscore
Scores=[]
Forsentenceinsentences:
  Words=sentence.split()
  Score=len(words)
  Scores[sentence]=score
```

```
#Sortthesentencesbasedontheirscores
Sorted_sentences=sorted(scores.items(),key=lambdax:[1],reverse=True)
```


```
#Extractthetop2sentenceswiththehighestscoresasthesummary
Summary_sentences=[sentence[0]forsentenceinsorted_sentences[:2]]
Summary=""
join(summary_sentences)
```

#Printthesummary
Print(summary)

@Slip-18

Q.1)WriteaJavaScriptProgramtoprintFibonacci numberononclievent. Ans:

```
<DOCTYPEhtml>
```

 Edit with WPS Office


```
<html>
<head>
  <title>FibonacciNumbers</title>
  <script>
    FunctiongenerateFibonacci(){
      //Gettheinputvaluefromtheuser
      Varinput=document.getElementById("inputNumber").value;
      Varoutput=document.getElementById("output");

      //Converttheinputtonumber
      Varn=parseInt(input);

      //Createanarraytostorethefibonacciquence
      Varfib=[];

      //Calculatethefibonacciquenceupton
      Fib[0]=0;
      Fib[1]=1;
      For(vari=2;i<=n;i++){
        Fib[i]=Fib[i-1]+Fib[i-2];
      }

      //Displaythefibonacciquence
      Output.innerHTML="Fibonacciquenceupto"+n+"*"+Fib.join(",");
    }
  </script>
</head> <body>
  <h1>FibonacciNumbers</h1>
  <p>Enteranumber:</p>
  <inputtype="text" id="inputNumber">
  <buttononclick="generateFibonacci()">GenerateFibonacci</button> <pid="output"></p>
</body>
</html>
```

 Edit with WPS Office

Q.2)Consideranytextparagraph.Removevestopwords.Tokenizetheparagraphtoextract wordsand Sentences.Calculatethewordfrequencydistributionandplotthefrequencies.Plotthewordcloudofthe Txt.

Ans:

```
#Installtheibraries
!pipinstallnltkmatplotlibwordcloud
```


```
#Importthenecessarymodules
Importnltk
Fromnltk.corpusimportstopwords
Fromnltk.tokenizeimportword_tokenize,sent_tokenize
Fromnltk.probabilityimportFreqDist
Importmatplotlib.pyplotaspilt
FromwordcloudimportWordCloud
```

```
#Downloadthetopwordscorpus
Nltk.download('stopwords')
```

```
#Definethetextparagraph
Text="Loremipsumdolorsitamet,consecteturadipiscingelit.Sedtristiquantevelit vestibulum,vulpharetraorciaculis.Nullamattisrisusquisauguetinciduntthonus.Morbi varius,arcuittascelerisquealio reet,magnaestimperdietquam,statmeturiceslectusjusto enim.Seddictumusscipitcommodo.Sednaemusconsequatrisus,necpharetrarnib interdumquis.Etiamegetquamvelauguedictumdignissimatanecelit.Nuncatapien dolor.Nullavitaetaculislorem.Suspendissepotenti.Sednonanteturpis.Morbiconsectetur, arcuavestibulumusscipit,mauriserosconvallisnib,necfeugiatocienimsitametenim. Aliquameratvolutpat.Etiamevelisidnequevevradapibusnonnoluctus."
```

```
#Tokenizetheparagraphtoextractwordssentences
Words=word_tokenize(text.lower())
Sentences=sent_tokenize(text)
```

#Removestopwordsfrotheextractedwords

 Edit with WPS Office

```
Stop_words=set(stopwords.words('english'))
Filtered_words=[wordforwordinwordsifword.casefold()notinstop_words]
```

```
#Calculatethewordfrequencydistributionandplotthefrequeciesusingmatplotlib
fdist=FreqDist(filtered_words)
fdist.plot(30,cumulative=False)
plt.show()
```

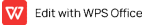
```
#Plotthewordcloudofthetextusingwordcloud
Wordcloud=WordCloud(width=800,height=800,
                    Background_color='white',
                    Stopwords=stop_words,
                    Min_font_size=10).generate(text)
#plottheWordCloudimage
plt.figure(figsize=(8,8),facecolor=None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
```

plt.show()

@Slip-19

Q.1)WriteaJavaScriptProgramtovalidateusernameandpasswordnononSubmitevent.
Ans:

```
<!DOCTYPEhtml>
<html>
<head>
<title>ValidateUserNameandPassword</title>
<script>
FunctionvalidateForm(){
```



Edit with WPS Office

```
Varusername=document.forms["myform"]["username"].value;
Varpassword=document.forms["myform"]["password"].value;
```

```
If(username==""){
Alert("Usernamemustbefilledout");
Returnfalse;
}

If(password==""){
Alert("Passwordmustbefilledout");
Returnfalse;
}
}
```

```
</script>
</head>
<body>
<h2>ValidateUserNameandPassword</h2>
<formname="myform"onsubmit="returnvalidateForm()"method="post">
<labelfor="username">Username:</label>
<inputtype="text" id="username" name="username"><br><br>
<labelfor="password">Password:</label>
<inputtype="password" id="password" name="password"><br><br>
<inputtype="submit" value="Submit">
</form>
</body>
</html>
```

Q.2)Downloadthemovie_review.csvdatasetfromKagglebyusingthefollowinglink
:https://www.kaggle.com/nltkdata/movie-review/version/3?select=movie_review.csvto perform
Sentimentalanalysisonabovedatasetandcreateawordcloud.

Ans:

```
<!DOCTYPEhtml>
<html>
<head>
<title>NumberCheck</title>
<scriptsrc="<?phpcheckboxbase_url('js/numbercheck.js');"></script>
</head>
<body>
<h1>NumberCheck</h1>
<p>Enteranumbertocheck:</p>
<inputtype="number" id="num"/>
<buttononclick="checkNumber(document.getElementById('num').value)">Check</button>
</body>
</html>
```

Createisfilechecknumber.js

```
FunctioncheckNumber(num){
If(num>0){
Alert("Thenumberispositive.");
}elseIf(num<0){
Alert("Thenumberisnegative.");
}else{
Alert("Thenumberiszero.");
}
}
```

Q.2)BuildasimplelinearegressionmodelforUserData.

Ans:
importpandasaspd
fromsklearn.model_selectionimporttrain_test_split
fromsklearn.linear_modelimportLinearRegression
fromsklearn.metricsimportmean_squared_error,r2_score

```
importpandasaspd
fromtextblobimportTextBlob
fromwordcloudimportWordCloud,STOPWORDS
importmatplotlib.pyplotasplt
!loadthedataset
Df=pd.read_csv('movie_review.csv')
```

```
#AddacolumnforsentimentanalysisusingTextBlob
Df['Sentiment']=df['Review'].apply(lambda x:TextBlob(x).sentiment.polarity)
```

#Createanewdataframeforpositivereviewsonly Pos_df=df[df['Sentiment']>0.2]

```
#Createawordcloudforpositivereviews
Wordcloud=WordCloud(width=800,height=800,
                    Background_color='white',
                    Stopwords=STOPWORDS,
                    Min_font_size=10).generate(" ".join(pos_df['Review']))
```

```
#Plotthewordcloud
plt.figure(figsize=(8,8),facecolor=None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)
```

plt.show()

@Slip-20

Q.1)createastudent.xmlfilecontainingateastStudentinformation.
Ans:
<?xmlversion="1.0"?>
<students>

```
importmatplotlib.pyplotasplt

#1.Collectdata
Data=pd.read_csv('user_data.csv')

#2.Preprocessdata
Data.dropna(inplace=True)
X=data['age'].values.reshape(-1,1)
Y=data['income'].values.reshape(-1,1)

#3.Splitdata
X_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

#4.Trainthemodel
Regressor=LinearRegression()
Regressor.fit(x_train,y_train)

#5.Predictvalues
Y_pred=Regressor.predict(x_test)

#6.Evaluatemodel
Mse=mean_squared_error(y_test,y_pred)
R2=r2_score(y_test,y_pred)
Print("Meansquarederror:"mse)
Print("R-squared:"r2)

#7.Visualizeresults
plt.scatter(x_test,y_test,color='gray')
plt.plot(x_test,y_pred,color='red',linewidth=2)
plt.show()
```

@Slip-22

Q.1)Createatablestudenthavingattributes(rollno,name,class).Usingcodeigniter,connectto the

```
<student>
<name>JohnDoe</name>
<age>21</age>
<gender>Male</gender>
<major>ComputerScience</major>
<gpa>3.8</gpa>
</student>
<student>
<name>JaneSmith</name>
<age>19</age>
<gender>Female</gender>
<major>Business</major>
<gpa>3.5</gpa>
</student>
<student>
<name>TomIhson</name>
<age>20</age>
<gender>Male</gender>
<major>Engineering</major>
<gpa>3.2</gpa>
</student>
<student>
<name>SaraLee</name>
<age>22</age>
<gender>Female</gender>
<major>Psychology</major>
<gpa>3.6</gpa>
</student>
<student>
<name>MikeBrown</name>
<age>18</age>
<gender>Male</gender>
<major>Education</major>
<gpa>3.4</gpa>
```

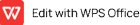
```
Databaseandinsertrecordsinit.
Ans:
<?php

//EstablishconnectiontoPostgreSQLdatabase
$conn=pg_connect("host=localhostdbname=your_database_nameuser=your_username password=your_password");

//Checkifconnectionwassuccessful
If($conn){
Echo"Connectionfailed.";
Exit;
}

//Createstudenttable
$query="CREATETABLEStudent(
RollnoINTEGERPRIMARYKEY,
NameVARCHAR(50)NOTNULL,
ClassVARCHAR(10)NOTNULL
)";
$result=pg_query($conn,$query);

If($result){
Echo"Errorcreatingtable: ".pg_last_error($conn);
Exit;
}else{
Echo"Tablecreatedsuccessfully.<br>";
}
//InsertRecordsintostudenttable
$insert_query="INSERTINTOstudent(rollno,name,class)
VALUES(1,'JohnDoe','10A'),
(2,'JaneSmith','9B'),
(3,'BobJohnson','11C'),
```



Edit with WPS Office

</student> </students>

Q.2)Considerthetextparagraph."""Helloai,WelcomeToPythonProgrammingAcademy.Python
ProgrammingAcademyisaniceplatformtolearnnewprogrammingskills.Itsdifficulttogetenrolled
inthisAcademy.""Removevestopwords.

Ans:

```
importnltk
fromnltk.corpusimportstopwords
nltk.download('stopwords')
```

```
#Textparagraph
Text="Helloai,WelcomeToPythonProgrammingAcademy.PythonProgrammingAcademyis  
aniceplatformtolearnnewprogrammingskills.ItsdifficulttogetenrolledinthisAcademy."
```

```
#Tokenizethetext
Tokens=nltk.word_tokenize(text)
!Removestopwords
Stop_words=set(stopwords.words('english'))
Filtered_tokens=[wordforwordintokensifnotword.lower()instop_words]
```

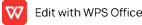
```
#Printthefilteredtokens
Print(filtered_tokens)
```

@Slip-21

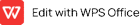
Q.1)AddaJavaScriptfileinCodeIgniter.Thejavascriptcodeshouldcheckwhetheranumber is
Positiveornegative.

Ans:

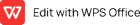
Htmlfile



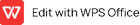
Edit with WPS Office



Edit with WPS Office



Edit with WPS Office



Edit with WPS Office

```
(4,'SarahLee','12D'),
(5,'TomBrown','8E');
```

```
$insert_result=pg_query($conn,$insert_query);
```

```
if(!$insert_result){
    Echo"Errorinsertingrecords:"._pg_last_error($conn);
    Exit;
}else{
    Echo"Recordsinsertedsuccessfully.";
}
```

```
//Closedatabaseconnection
```

```
Pg_close($conn);
```

```
>}
```

Q2) Consider any text paragraph. Remove the stop words.

Ans:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import import word_tokenize
```

```
#sample text paragraph
Text="Helloai, WelcometoPythonProgrammingAcademy.PythonProgrammingAcademyis
aniceplatformtolearnnewprogrammingskills.ItsdifficulttogetenrolledinthisAcademy."
```

```
#tokenize the text paragraph
Words=word_tokenize(text)
```

```
#define stop words
```



Edit with WPS Office

```
Stop_words=set(stopwords.words('english'))
```

```
#removestop words
Filtered_words=[wordforwordinwordifword.casefold()notinstop_words]
```

```
#join filtered words to form a sentence
Filtered_sentence=" ".join(Filtered_words)
```

```
Print(Filtered_sentence)
```

```
@Slip-23
```

Q.1) Create a table student having attributes (rollno, name, class) containing at least 5 records. Using Codeigniter, display all its records.

Ans:

```
<?php
```

```
//Establish connection to PostgreSQL database
```

```
$conn=pg_connect("host=localhost dbname=your_database nameuser=your_username password=your_password");
```

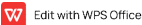
```
//Check if connection was successful
```

```
if(!$conn){
    Echo"Connection failed.";
    Exit;
```

```
}
```

```
//Create student table
```

```
$query="CREATE TABLE student(
    Rollno INTEGER PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Class VARCHAR(10) NOT NULL
```



Edit with WPS Office

```
);
```

```
$result=pg_query($conn,$query);
```

```
if(!$result){
    Echo"Error creating table:"._pg_last_error($conn);
    Exit;
}else{
    Echo"Table created successfully.<br>";
}
```

```
//Insert 5 records into student table
```

```
$insert_query="INSERT INTO student(rollno,name,class) VALUES(1,'JohnDoe','10A'),
(2,'JaneSmith','9B'),
(3,'BobJohnson','11C'),
(4,'SarahLee','12D'),
(5,'TomBrown','8E');
```

```
$insert_result=pg_query($conn,$insert_query);
```

```
if(!$insert_result){
    Echo"Error inserting records:"._pg_last_error($conn); Exit;
```

```
}else{
    Echo"Records inserted successfully.";
}
```

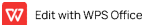
```
//Closedatabaseconnection
```

```
Pg_close($conn);
```

```
//function to display database records
```

```
function display_records($table_name){
```

```
//establish connection to PostgreSQL database
```



Edit with WPS Office

```
$conn=pg_connect("host=localhost dbname=your_database nameuser=your_username password=your_password");
```

```
//check if connection was successful
if(!$conn){
```

```
    Echo"Connection failed.";
    Exit;
}
```

```
//retrieve records from specified table
```

```
$query="SELECT * FROM ".$table_name; $result=pg_query($conn,$query);
```

```
//check if query was successful
```

```
if(!$result){
    Echo"Error retrieving records:"._pg_last_error($conn); Exit;
```

```
//display records in an HTML table
```

```
Echo"<table>";
Echo"<tr><th>RollNo</th><th>Name</th><th>Class</th></tr>";
While($row=pg_fetch_assoc($result)){
    Echo"<tr><td>".$row['rollno']."</td><td>".$row['name']."</td><td>".$row['class']."</td></tr>";
}
Echo"</table>";
```

```
//closedatabaseconnection Pg_close($conn);
```

```
}
```

```
>}
```

Q2) Consider any text paragraph. Preprocess the text to remove any special characters and Digits.

Ans:

```
Import re
```



Edit with WPS Office

Text="Hello, #world123! This is a sample text paragraph. It contains special characters and 5 digits."

```
#Remove special characters and digits
```

```
Processed_text=re.sub(r"[^a-zA-Z\W]"," ",text)
```

```
Print(processed_text)
```

```
@Slip-24
```

Q.1) Write a PHP script to create a student.xml file which contains student rollno, name, address, college and course. Print student details of specific course in a table format after accepting course as input. Ans:

```
<?php
```

```
//Define student details
```

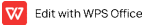
```
$students=array(
    Array("rollno"=>1,"name"=>"JohnDoe","address"=>"123MainSt","college"=>"ABC College","course"=>"ComputerScience"),
    Array("rollno"=>2,"name"=>"JaneSmith","address"=>"456MainSt","college"=>"DEF College","course"=>"InformationTechnology"),
    Array("rollno"=>3,"name"=>"BobJohnson","address"=>"789MainSt","college"=>"GHI College","course"=>"BusinessAdministration"),
    Array("rollno"=>4,"name"=>"SarahLee","address"=>"101MainSt","college"=>"JKL College","course"=>"Marketing"),
    Array("rollno"=>5,"name"=>"TomBrown","address"=>"121MainSt","college"=>"MNO College","course"=>"ComputerScience");
```

```
//Create a SimpleXMLElement object
```

```
$xml=new SimpleXMLElement('<students></students>');
```

```
//Add student elements to the XML object
```

```
foreach($students as $student){
```



Edit with WPS Office

```
$student_element=$xml->addChild('student');
$student_element->addChild('rollno',$student['rollno']); $student_element->addChild('name',$student['name']);

$student_element->addChild('address',$student['address']);
$student_element->addChild('college',$student['college']);
$student_element->addChild('course',$student['course']);
}
```

```
//Save the XML data to a file
```

```
$xml->asXML('student.xml');
```

```
//Get course input from user
```

```
$course=isset($_POST['course'])?$_POST['course']:"";
```

```
//Load the XML file
```

```
$xml=simplexml_load_file('student.xml');
//Find students with matching course
$filtered_students=$xml->xpath("//student[course='".$course."']");
```

```
//Print table of matching students
```

```
Echo"<table border='1'>";
Echo"<tr><th>Roll No</th><th>Name</th><th>Address</th><th>College</th><th>Course</th></tr>";
foreach($filtered_students as $student){
    Echo"<tr>";
    Echo"<td>{$student->rollno}</td>"; Echo"<td>{$student->name}</td>";

    Echo"<td>{$student->address}</td>";
    Echo"<td>{$student->college}</td>";
    Echo"<td>{$student->course}</td>";
    Echo"</tr>";
}
Echo"</table>";
?>
```

Q.2) Consider the following dataset:
https://www.kaggle.com/datasets/datasnaek/youtubenew?select=InVideos.csv
Write a Python script for the following: i.

Read the dataset and perform data cleaning operations on it. ii.
ii. Find the total views, total likes, total dislikes and comment count.

Ans:

```
import pandas as pd
```

```
#Read the dataset
```

```
Df=pd.read_csv('InVideos.csv')
```

```
#Drop the columns that are not required
```

```
Df=Df.drop(['video_id','trending_date','channel_title','category_id','publish_time','tags',
'thumbnail_link','comments_disabled','ratings_disabled','video_error_or_removed'],axis=1)
```

```
#Convert the data type of 'views','likes','dislikes', and 'comment_count' to integer
```

```
Df['views','likes','dislikes','comment_count']=Df['views','likes','dislikes','comment_count'].astype(int)
```

```
#Find the total views, likes, dislikes, and comment count
```

```
Total_views=Df['views'].sum()
Total_likes=Df['likes'].sum()
Total_dislikes=Df['dislikes'].sum()
Total_comments=Df['comment_count'].sum()
```

```
Print('Total Views:',total_views)
```

```
Print('Total Likes:',total_likes)
```

```
Print('Total Dislikes:',total_dislikes)
```

```
Print('Total Comments:',total_comments)
@Slip-25
```

Q.1) Write a script to create 'cricket.xml' file with multiple elements as shown below:

```
<CricketTeam>
```

```
<Team country="Australia">
```

```
<player>____</player>
```

```
<runs>____</runs>
```

```
<wicket>____</wicket>
```

```
</Team>
```

```
</CricketTeam>
```

```
Write a script to add multiple elements in "cricket.xml" file of category, country="India".
```

Ans:

```
<?php
```

```
//Create a new DOM document
```

```
$doc=new DOMDocument();
```

```
//Create the root element
```

```
$cricketTeam=$doc->createElement("CricketTeam");
```

```
//Create the first team element for Australia
```

```
$teamAustralia=$doc->createElement("Team");
```

```
$teamAustralia->setAttribute("country","Australia");
```

```
//Create the player element and set its value
```

```
$player1=$doc->createElement("player","SteveSmith");
```

```
$teamAustralia->appendChild($player1);
```

```
//Create the run element and set its value
```

```
$runs1=$doc->createElement("runs","7090");
```

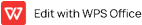
```
$teamAustralia->appendChild($runs1);
```

```
//Create the wicket element and set its value
```

```
$wicket1=$doc->createElement("wicket","17");
```

```
$teamAustralia->appendChild($wicket1);
```

```
//Append the team element to the root element $cricketTeam->appendChild($teamAustralia);
```



Edit with WPS Office


```
//CreatethesecondelementforIndia
$teamIndia->$doc->createElement("Team");
$teamIndia->setAttribute("country","India");

//Createtheplayerelementandsetitsvalue
$player2->$doc->createElement("player","ViratKohli");
$teamIndia->appendChild($player2);
```

```
//Createtherunselementandsetitsvalue
$runs2->$doc->createElement("runs","12169");
$teamIndia->appendChild($runs2);
```

```
//Createthewicketelementandsetitsvalue
$wicket2->$doc->createElement("wicket","4");
$teamIndia->appendChild($wicket2);
```

```
//Createthecategoryelementandsetitsvalue
$category->$doc->createElement("category","Captain");
$teamIndia->appendChild($category);
```

```
//Appendtheteamelementtotherootelement
$cricketTeam->appendChild($teamIndia);
```

```
//Appendtherootelementtothedocument
$doc->appendChild($cricketTeam);
```

```
//SavetheXMLfile
$doc->save("cricket.xml");
```

```
Echo"Elementsaddedsuccessfully!";
?>
```

Q.2)Considerthefollowingdataset:https://www.kaggle.com/datasets/seunguini/youtubecommentsfor-covid19-relatedvideos?select=covid_2021_1.csv WriteaPythonscriptforthefollowing: i.

Readthedataandperformdatacleaningoperationsonit. ii.

Tokenizethecommentsinwords.iii.Performsentimentanalysisandfindthepcentageof positive,negativeandneutralcomments..

Ans:

```
import pandas as pd
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```


```
#read the dataset
Df=pd.read_csv('covid_2021_1.csv')
```

```
#remove null values and duplicates
Df.dropna(inplace=True)
Df.drop_duplicates(subset='Comment',inplace=True)
```

```
#tokenize comments in words
nltk.download('punkt')
Df['tokens']=Df['Comment'].apply(nltk.word_tokenize)
```

```
#perform sentiment analysis
nltk.download('vader_lexicon')
sia=SentimentIntensityAnalyzer()
Df['sentiment']=Df['Comment'].apply(lambda x:sia.polarity_scores(x)['compound'])
```

```
#calculate percentage of positive,negative,and neutral comments
Total_comments=len(Df)
Positive_comments=len(Df[Df['sentiment']>0])
Negative_comments=len(Df[Df['sentiment']<0])
Neutral_comments=len(Df[Df['sentiment']==0])
```

 Edit with WPS Office

```
Positive_percentage=(positive_comments/total_comments)*100
Negative_percentage=(negative_comments/total_comments)*100
Neutral_percentage=(neutral_comments/total_comments)*100
```

```
#print the results
Print('TotalComments:',total_comments)
Print('PositiveComments:',positive_comments,','positive_percentage,%)')
Print('NegativeComments:',negative_comments,','negative_percentage,%)')
Print('NeutralComments:',neutral_comments,','neutral_percentage,%)')
```

@Slip-26

```
Q.1)Create an employee table as follows EMP(eno,ename,designation,salary). Write Ajax program to select the employee's name and print the selected employee's details.
```

Ans:


Html file

```
<select id="employee-list">
<option value="">Select an employee</option>
<!--Populate the dropdown with employee names using PHP-->
</select>
```

```
<div id="employee-details">
<!--Employee details will be displayed here-->
</div>
```

Ajax file

```
$(document).ready(function(){
```

 Edit with WPS Office

```
//Add event listener to the select dropdown
$( '#employee-list' ).change(function(){
    var selectedEmployee = $(this).val();
    //Make an AJAX request to fetch employee details
    $.ajax({ url: 'empdetails.php', type: 'POST',
        data: { employeeName: selectedEmployee }, dataType: 'json',
        success: function(response){

            //Parse the JSON response and display employee details
            var detailsHtml = 'Employee Name: '+ response.ename + '<br>' +
                'Designation: '+ response.designation + '<br>' +
```


```
                'Salary: '+ response.salary;
            $('#employee-details').html(detailsHtml);
        }
    });
});
});
```

Php file empdetails.php

```
<?php
//Establish database connection
$conn = pg_connect("host=localhost dbname=database name user=username password=password");
if($conn){
    die("Connection failed: " . pg_last_error());
}
```

```
//Get the selected employee name from AJAX request
$employeeName = $_POST['employeeName'];
```

//Query the EMP table for the details of the selected employee

 Edit with WPS Office

```
$sql="SELECT * FROM EMP WHERE ename='EmployeeName'";
$result=pg_query($conn,$sql);
```


```
if(pg_num_rows($result)>0){
    //Build a JSON object with employee details
    $employee=pg_fetch_assoc($result);
    $response=array(
        'ename'=>$employee['ename'],
        'designation'=>$employee['designation'],
        'salary'=>$employee['salary']
    );
    echo json_encode($response);
}else{
    Echo "Employee not found";
}
```

```
//Close database connection
pg_close($conn);
?>
Q.2)Consider the text paragraph: ""Hello!ai, Welcometo Python Programming Academy. Python Programming Academy is an e-platform to learn new programming skills. It is difficult to get enrolled in this Academy."" Preprocess the text to remove any special characters and digits. Generate the Summary using extractive summarization process.Q.
```

Ans:

```
import re
from nltk.tokenize import sent_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```
#Text to summarize
Text="Hello!ai, Welcometo Python Programming Academy. Python Programming Academy is an e-platform to learn new programming skills. It is difficult to get enrolled in this Academy."
```

 Edit with WPS Office

```
#Preprocess the text to remove special characters and digits
Preprocessed_text=re.sub(r"[^a-zA-Z\s]","",text)
```

```
#Tokenize the preprocessed text into sentences
Sentences=sent_tokenize(preprocessed_text)
```

```
#Calculate the importance score for each sentence using TF-IDF
Vectorizer=TfidfVectorizer()
Tfidf_matrix=Vectorizer.fit_transform(Sentences)
Similarity_matrix=cosine_similarity(Tfidf_matrix)
```

```
#Select top N sentences based on their importance score
N=2
Top_sentences=sorted(range(len(similarity_matrix)-1),key=lambda i:similarity_matrix[-1][i])[-N:]
```

```
#Concatenate the top sentences to form the summary
Summary=""
for i in top_sentences:
    Summary+=sentences[i]+" "
```

Print(summary)

@Slip-27


```
Q.1)Create a web application that contains voters' details and check proper validation for (name, Age, and nationality), as names should be in uppercase letters only, Ages should not be less than 18 yrs and Nationality should be Indian. (use HTML-AJAX-PHP).
```

Ans:

Html file

```
<!DOCTYPE html>
<html>
<head>
    <title>VoterDetails</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</head>
<body>
    <h2>VoterDetails</h2>
    <form id="voterForm">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required><br>
        <label for="age">Age:</label>
        <input type="number" id="age" name="age" required><br>
        <label for="nationality">Nationality:</label>
        <input type="text" id="nationality" name="nationality" required><br>
        <input type="submit" value="Submit">
    </form>
    <div id="response"></div>
</script>

$(document).ready(function(){
    $('#voterForm').submit(function(event){
        event.preventDefault();
        var name=$('#name').val().toUpperCase();
        var age=$('#age').val();
        var nationality=$('#nationality').val();
        $.ajax({ url:'voter.php', method:'POST',
            data:{name:name,age:age,nationality:nationality},
            success:function(response){
                $('#response').html(response);
            }
        });
    });
});
</script>
</body>
```

 Edit with WPS Office

</html>

Voter.php file

```
<?php
$name=$_POST['name'];
$age=$_POST['age'];
$nationality=$_POST['nationality'];
```

```
if(preg_match('/^[A-Z]$/',$name)){
    Echo"Names should be in uppercase letters only.";
}else if($age<18){
    Echo"Ages should not be less than 18 years.";
}else if(strcasecmp($nationality,'Indian')==0){
    Echo"Nationality should be Indian.";
```


```

}else{
    Echo"Validations successful.Voter details:<br>Name: ".$name."<br>Age: ".$age."<br>Nationality: ".$nationality;
}
?>
```

Q.2)Create your own transactions data set and apply the above process on your dataset Ans:

```
import random
import csv
```

```
#Generate random transaction data
Transactions=[]
for i in range(1,101):
    Transaction_id=i
    Transaction_date="2022-05-"+random.randint(1,31)+"/026"
    Customer_id=random.randint(1,10)
    Item_id=random.choice(["A","B","C","I"])
```

 Edit with WPS Office

```
Item_price=round(random.uniform(10.0,100.0),2)
Quantity=random.randint(1,10)
Transactions.append([transaction_id,transaction_date,customer_id,item_id,item_price, quantity])

#SavethedataaCSVfile
Withopen('transactions.csv','w','newline="")ascsvfile:
    Writer=csv.writer(csvfile)
    Writer.writerow(["TransactionID","TransactionDate","CustomerID","ItemID","ItemPrice",
"Quantity"])
    Fortransactionintransactions: Writer.writerow(transaction)

Importpandasaspd

#ReadtheCSVfileintoaPandasDataFrame Df=pd.read_csv('transactions.csv')

#Convertthe"ItemPrice"column tonumerictype Df['ItemPrice']=pd.to_numeric(Df['ItemPrice'])
```

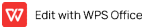
```
#Calculatethesalesamountforeachtransaction
Df['Sales']=df['ItemPrice']*df['Quantity'] #GrouphetransactionsbycustomerDandcalculatethetotalsalesforeachcustomer
Total_sales=df.groupby('CustomerID')['Sales'].sum().reset_index()
```

```
#Printtheresults Print(total_sales)
```

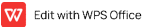
@Slip-28

Q.1)WriteaPHPscriptusingAJAXconcept,tocheckusernameandpasswordarevalidor invalid(use Databasestoreusernameandpassword).

Ans:



```
Htmlfile
<DOCTYPEhtml>
<html>
<head>
    <title>Login</title>
    <scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script> <script>
        $(document).ready(function(){
            $("*Login-form").submit(function(event){
                Event.preventDefault();
                Varusername=$("#username").val();
                Varpassword=$("#password").val();
                $.ajax({ url:'check_login.php', type:'post',
                    data:{username:username,password:password},
                    success:function(response){ if(response=="valid"){
                        window.location.href="dashboard.php";
                    }
                    Else{
                        Alert("Invalidusernameorpassword");
                    }
                }
            });
        });
    </script>
</head><body>
    <h2>Login</h2>
    <formid="login-form"method="post">
        <label>Username:</label>
        <inputtype="text"name="username"id="username"><br><br>
        <label>Password:</label>
```



```
<inputtype="password"name="password"id="password"><br><br>
<inputtype="submit"value="Login">
</form>
</body>
</html>

Phpfilesccheck_login.php

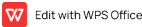
<?php
//Establishdatabaseconnection
$conn=mysqli_connect('localhost','username','password','database_name');
if($conn){
    Die('Connectionfailed:'.mysqli_connect_error());
}

//GetusernameandpasswordfromAJAXrequest
$username=$_POST['username'];
$password=$_POST['password'];

//Querytheuserstableforthecorrectusernameandpassword
$sql="SELECT*FROMUsersWHEREusername='$username'ANDpassword='$password'";
$result=mysqli_query($conn,$sql);

if(mysqli_num_rows($result)>0){
    Echo"valid";
}else{
    Echo"invalid";
}

//Closedatabaseconnection
mysqli_close($conn);
?>
```



Q.2)BuildasimplelinearregrressionmodelforCarDataset.

Ans:

```
Fromsklearn.linear_modelimportLinearRegression
```

```
Mileage=[10],[20],[30],[40],[50],[60],[70],[80] Price=[24,19,17,13,10,7,5,2]
```

```
Reg=LinearRegression().fit(mileage,price)
```

```
Print("Intercept",reg.intercept_)
Print("Coefficient",reg.coef_[0])
```

```
New_mileage=[[25],[45],[65]]
Predicted_price=reg.predict(new_mileage)
```

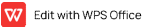
```
Print("Predictedprices:",predicted_price)
```

@Slip-29

Q.1)WriteaPHPscriptforthe following:Designaformtoacceptanumberfromtheuser. Performtheoperationsandshowtheresults.

- 1)FibonacciSeries.
 - 2)Tofindsumofthedigitsofthatnumber. (Use theconceptofselfprocessingpage.)
- Ans:

```
<DOCTYPEhtml>
<html>
<head>
    <title>NumberOperations</title>
```



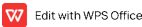
```
</head><body>
    <h1>NumberOperations</h1>
    <?php
        //definevariablesandsettoemptyvalues $num=$op="";

        if($_SERVER["REQUEST_METHOD"]=="POST"){
            $num=test_input($_POST["num"]);
            $op=test_input($_POST["op"]);

            //performoperationbasedonuser'schoicse
            Switch($op){
                Case"fib":
                    $result=fibonacci($num);
                    Echo"<p>TheFibonacciSeriesof$Numnumbersis:$result</p>"; Break; Case"sum":
                        $result=sumOfDigits($num);
                        Echo"<p>Thesumofdigitsin$Numis:$result</p>"; Break; Default:
                            Echo"<p>Invalidoperationselected</p>";
            }
        }

        Functiontest_input($data){
            $data=trim($data);
            $data=stripslashes($data);
            $data=htmlspecialchrs($data);
            Return$data;
        }

        Functionfibonacci($num){
            $first=0;
            $second=1; $result="";
```



```
For($i=0;$i<$num;$i++){
    $result+=$first."";
    $third=$first+$second;
    $first=$second;
    $second=$third;
}

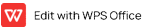
Return$result;

FunctionsumOfDigits($num){
    $sum=0;
    While($num>0){
        $digit=$num%10;
        $sum+=$digit;
        $num=(int)($num/10);
    }

    Return$sum;
}

?>

<formmethod="post"action="<?phpecho htmlspecialchars($_SERVER["PHP_SELF"]);?>"
    <labelform="num">Enteranumber:</label>
    <inputtype="number"name="num"id="num"required>
    <br><br>
    <labelform="op">Selectanoperation:</label>
    <selectname="op"id="op"required>
        <optionvalue="">--Select--</option>
        <optionvalue="fib">FibonacciSeries</option>
        <optionvalue="sum">SumofDigits</option>
    </select>
<br><br>
```



```
<inputtype="submit"value="Submit">
</form>
</body>
</html>

Q.2)BuildalogisticregressionmodelforStudentScoreDataset.
Ans:

#Importnecessarylibraries
Importpandasaspd
Fromsklearn.linear_modelimportLogisticRegression
Fromsklearn.model_selectionimporttrain_test_split
Fromsklearn.metricsimportaccuracy_score

#Loadthedataset
Data=pd.read_csv('student_scores.csv')

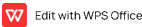
#Splithedataintoinputandoutputvariables
X=data.iloc[:,:-1].values
Y=data.iloc[:, -1].values

#Splithedataintotrainingandtestingsets
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)

#Createthelogisticregressionmodelandfittothetrainingdata
Classifier=LogisticRegression() Classifier.fit(X_train,y_train)

#MakepredictionsonthetestingsetY_pred=classifier.predict(X_test)

#Evaluatethemodel'saccuracy
Accuracy=accuracy_score(Y_test,Y_pred)
```



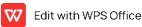
```
Print("Accuracy:",accuracy)
@Slip-30
```

Q.1)CreateaXMLfilewhichgivesdetailsofbooksavailablein "Bookstore"fromfollowing Categories.

- 1)Yoga
 - 2)Story
 - 3)Technical
- Andelementsin eachcategoryareinthefollowingformat
- ```
<Book>
<Book_Title>
-----</Book_Title>
<Book_Author>-----</Book_Author>
<Book_Price>
-----</Book_Price>
</Book>
Savethefileas"Bookcategory.xml"
```

Ans:

```
<?xmlve="?"xmlversion="1.0"encoding="UTF-8"?>
<Bookstore>
 <Yoga>
 <Book>
 <Book_Title>LightonYoga</Book_Title>
 <Book_Author>B.K.S. Iyengar</Book_Author>
 <Book_Price>20.99</Book_Price>
 </Book>
 </Yoga>
 <Book>
 <Book_Title>TheYogaBible</Book_Title>
 <Book_Author>ChristinaBrown</Book_Author>
 <Book_Price>15.50</Book_Price>
```



```
</Book>
</Yoga>
<Story>
<Book>
 <Book_Title>The Alchemist</Book_Title>
 <Book_Author>Paulo Coelho</Book_Author>
 <Book_Price>12.99</Book_Price>
</Book>
<Book>
 <Book_Title>The Da Vinci Code</Book_Title>
 <Book_Author>Dan Brown</Book_Author>
 <Book_Price>14.75</Book_Price>
</Book>
</Story>
<Technical>
<Book>
 <Book_Title>Python for Data Science Handbook</Book_Title>
 <Book_Author>Jake VanderPlas</Book_Author>
 <Book_Price>28.99</Book_Price>
</Book>
<Book>
 <Book_Title>Cracking the Coding Interview</Book_Title>
 <Book_Author>Gayle Laakmann McDowell</Book_Author>
 <Book_Price>23.50</Book_Price>
</Book>
</Technical>
</Bookstore>
```

```
Q.2) Create the dataset.transactions=[['eggs','milk','bread'],['eggs','apple'],['milk','bread'],['apple',
'milk'],['milk','apple','bread']].
```

Convert the categorical values into numeric format. Apply the apriori algorithm on the above dataset to

Generate the frequent itemsets and association rules.

Ans:

```
Transactions=[['eggs','milk','bread'],['eggs','apple'],['milk','bread'],['apple','milk'],['milk','apple','bread']]
```

```
Create a dictionary to map itemset to unique numeric values
Item_to_num={'eggs':1,'milk':2,'bread':3,'apple':4}
```

# Convert the categorical values in the dataset to numeric values

```
Numeric_transactions=[]
```

For transaction in transactions:

```
 Numeric_transaction=[Item_to_num[item] for item in transaction]
```

```
 Numeric_transactions.append(Numeric_transaction)
```

```
Print(Numeric_transactions)
from mlxtend.frequent_patterns import apriori, association_rules
```

# Generate frequent itemsets with a minimum support of 0.4

```
Frequent_itemsets=apriori(Numeric_transactions,min_support=0.4,use_colnames=True)
```

# Generate association rules with a minimum confidence of 0.7

```
Rules=association_rules(Frequent_itemsets,metric="confidence",min_threshold=0.7)
```

```
Print(Frequent_itemsets)
```

```
Print(Rules)
```